Investigating the Influence of Inspector Learning Styles on Design Inspections: Findings of a Quasi-Experiment

Gisele Carneiro[1], Rodrigo Laigner[1], Marcos Kalinowski[1],
Dietmar Winkler[2], Stefan Biffl[2]

[1] Computing Institute, Fluminense Federal University, Niterói, Brazil
gcarneiro@ic.uff.br, rodrigo.laigner@id.uff.br, kalinowski@ic.uff.br

[2] Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria
dietmar.winkler@tuwien.ac.at, stefan.biffl@tuwien.ac.at

Abstract. Software inspections are an efficient mean to improve quality. Learning Styles (LS) have been used to detect an individual's preferences to acquire and process information according to different dimensions. For requirements inspections, inspection teams with different LSs showed more effective at detecting defects than teams with similar LS. The goal of this paper is to investigate the influence of inspector's LS on the effectiveness and efficiency of design inspections. We conducted a quasi-experiment using a subset of a real requirements document and corresponding design diagrams with seeded defects, characterizing each of the participants according to their LSs. We also implemented a script to combine inspectors into nominal inspection teams. We observed that, for design inspections, teams with similar LSs were more effective and efficient than teams with dissimilar LSs. Design inspections seem to be affected more by other factors, such as defect detection techniques, than from selecting inspectors based on their learning styles. However, replications are needed to reinforce our results.

Keywords: software design inspection; inspection teams; learning styles; quasi-experiment.

1 Introduction

Software inspection [9] is a particular type of review with a rigorous and well-defined process. The importance of software inspections has been well documented [1,21]. According to Boehm and Basili [7], software inspections are able to catch about 60% of the defects and their benefit is increased when applying them in early development phases (e.g., during the requirements and design phase), when fixing the defects is considerably cheaper. Thus, they have proven an efficient means to improve quality and to reduce rework cost and future maintenance costs [3]. Inspections can be applied to any software artifact, such as requirements [23] and design documents [25].

Typical software inspections include individual inspection activities and team meetings to aggregate individual defect reports to a team defect report [3]. Thus, there

is a need to support team composition for inspection. However, real team meetings require additional resources. Johnson [16] and Votta [26] concluded that real meetings often have little effect on defect detection results but require considerably more effort. In this paper, we focus on the concept of nominal (non-communicating) teams and investigate the number of unique defects found by different team compositions.

Concerning the combination of the inspectors, many characteristics (e.g., domain experience or inspection experience) could be used for this purpose. In this study, we address one particular characterization, which has been recently used in studies on combining inspectors for requirements inspections [12,13,14,15]: their underlying Learning Styles (LS) [10]. LSs have been used to measure an individual's characteristic to acquire and process information [10]. In fact, during design inspections it is necessary to acquire information from the requirements document and to process this information when analyzing the design documents. The main goal of this study is to investigate the influence of LS on inspector effectiveness and efficiency, individually and when combining them into teams.

To address this goal, we conducted a quasi-experiment based on a set of real-world requirements and design documents. Our study focuses on design inspections with defect based reading technique. This technique allows the inspectors use the list of defect types to find the defects. The first trial of our experiment was conducted with 11 participants and allowed us to observe preliminary results indicating that specific LS seem to be more effective than others. Motivated by this, we conducted a second trial with additional 26 participants. From the aggregated results of the both trials we analyzed 23 results, this will be explained in section 5. The results reinforced our initial perception that specific learning styles seem to be more effective and efficient than others. Furthermore, for design inspections a variety of LSs does not seem to improve the effectiveness and efficiency of design inspection teams.

This paper is organized as follows: Section 2 describes the background and related work. Section 3 introduces to the research goal and presents research issues and Section 4 concerns the experiment design. In Section 5 the results are presented and analyzed. Threats to validity are discussed in Section 6. Finally, Section 7 concludes and identifies future work.

2 Background and Related Work

In this section, we describe background and related work on software inspections, learning styles, and empirical studies in this context.

2.1 Software Inspections

Software inspections improve software product quality by the analysis of software artifacts, detecting defects for removal before these artifacts are delivered to later software life cycle activities [7]. The traditional inspection process by Fagan [9] involves a moderator planning the inspection, inspectors reviewing the artifact, a team meeting to discuss and register defects, passing the defects to the author for rework, so

they can be corrected, and a final follow-up evaluation by the moderator on the need of a new inspection (i.e., a re-inspection [4]).

Over the years, many contributions on software inspections have been proposed, including alternative processes, techniques to improve the effectiveness and efficiency of inspectors in detecting defects, models and guidelines supporting tasks of the inspection process that involve decision making, as well as tool support [18]. Much knowledge on those contributions has been acquired by conducting empirical studies and some effort has been recently invested into organizing such knowledge into a body of knowledge for the area [5,6,22]. However, we found no studies (cf. Section 2.3) specifically addressing the influence of LS on design inspection effectiveness and efficiency, which is the focus of our paper.

2.2 Learning Styles

The ways in which an individual acquires, retains, and retrieves information have been collectively termed as the individual's Learning Style (LS) [19, 10]. Research results of LS in psychology, analyzing the LS of engineering students prove that an individual perceives and processes information better if the presentation is compliant with their preferred style [10]. In this paper, we focus on the model presented by Felder and Silverman, which was inspired by Jung [17] and Kolb [19] models. We choose this model because of the reliability [11][8] and good acceptance of participants when compared with other models [20]. The four fundamental LS dimensions as described by Felder and Silverman [10] are:

- *Active (ACT) vs. Reflective (REF).* While *active* learners prefer learning through active experimentation and by working in groups, i.e. to "try things out" and doing something with the information (e.g., discuss or test it). *Reflective* learners, on the other hand, prefer learning through reflective observation and working alone, i.e. to "observe things". *Active* learners tend to be experimentalists; *reflective* learners tend to be theoreticians.
- *Sensitive (SEN) vs. Intuitive (INT). Sensitive* learners are oriented toward facts – they prefer concrete data, focus on details, and follow existing ways. They dislike innovative methods. *Intuitive* learners, on the other hand are abstract, conceptual, innovative, oriented towards theories and meaning and discovering possibilities, they dislike repetition.
- *Visual (VIS) vs. Verbal (VRB). Visual* learners prefer visual representation of material (e.g., diagrams, video, or flow charts) while *verbal* learners prefer written/spoken explanations.
- *Sequential (SEQ) vs. Global (GLO). Sequential* learners are linear, orderly, and need to analyze situations step-by-step; *global* learners like to see information from a more global perspective.

To characterize individuals according to these LS dimensions, researchers from the North Carolina State University created the Index of Learning Styles (ILS)

questionnaire, which is a self-scoring instrument available online[1]. The ILS consists of 44 questions and provides a result characterizing an individual based on her/his answers in each of the four LS dimensions. Fig. 1 presents an example result of an ILS application.

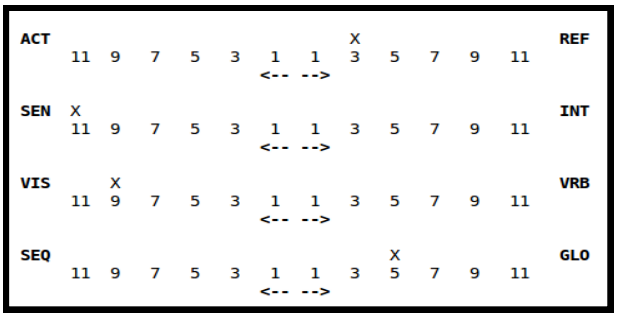

Fig. 1 Example Results of an ILS Questionnaire Application.

According to Fig. 1, an individual receives a score in each of the four LS dimensions. This score for each dimension can be 1, 3, 5, 7, 9 or 11 in the direction of the individual's preferred learning style. The LS of the individual characterized in the figure (one of the experiment participants) tends towards *reflective*, *sensitive*, *visual* and *global*.

2.3 Empirical Studies on Inspections and Learning Styles

The combination of inspectors by their LS characteristics has been recently investigated by Goswami *et al.* [12,13,14,15] in the context of requirements inspections. Among their findings, they concluded that, for requirements inspections, teams with diverse LSs were more effective and efficient in detecting defects when compared to teams of inspectors with similar LSs [12,13,14,15]. These studies motivated us to further investigate the impact of LSs on the effectiveness and efficiency when considering design inspection teams, where inspectors review documents including diagrams, i.e. graphical representations of concepts based on the requirements document. According to Felder and Silverman [10], the way information is acquired and processed is related to the LS (cf. Section 2.2). Therefore, we would like to know how the LS behave in the context acquiring information from requirements documents and processing such information in the context of reviewing software designs in terms of software engineering diagrams.

3 Study Goal and Research Questions

The goal of the study was to investigate the impact of inspector LS on design inspection effectiveness and efficiency. In particular, we were interested in identify the

1 Index of Learning Styles Questionnaire: http://www.engr.ncsu.edu/learningstyles/ilsweb

impact of LSs and whether inspection teams with diverse LSs would be able to detect more defects in design documents when compared to inspection teams with similar LSs. Table 1 presents this goal using in the GQM (*Goal Question Metric* paradigm) template [2].

Table 1. Goal of the Empirical Study.

Analyze	the influence of learning styles
For the purpose of	characterization
with respect to	individual and team effectiveness and efficiency
from the point of view of the	researchers
in the context of	novice inspectors, inspecting a real-world design document based on a real-world requirements document.

This study evaluated the effectiveness and efficiency of inspectors individually and for all potential inspection team combinations, considering different team sizes. The definitions used for effectiveness and efficiency are the same used in [3]:

- **Effectiveness** is the ratio between the number of defects found and the total of (seeded) defects in the documents.
- **Efficiency** refers to the number of defects found and the time spent in finding them, i.e., per time interval.

Based on the experiment goal, we defined two main research questions (RQs):

RQ-1: Do the LS dimensions have an impact on individual defect detection effectiveness and efficiency? As first step, i.e., before combining inspectors in teams, we wanted to observe the impact of the LS dimensions on individual defect detection performance. The main question focuses on the observation, if inspectors with a specific LS would achieve better results when reviewing the design diagrams based on the requirements. Note that in previous psychology studies engineering students proved to process information better if it is presented in their preferred LS [10].

To investigate this research questions, we formulated the following null and alternative hypotheses concerning the effectiveness and efficiency:

Hypothesis 01: LS dimensions have no influence on individual defect detection effectiveness.

Hypothesis A1: LS dimensions have influence on individual defect detection effectiveness.

Hypothesis 02: LS dimensions have no influence on individual defect detection efficiency.

Hypothesis A2: LS dimensions have influence on individual defect detection efficiency.

RQ-2: Looking at LS combinations, are heterogeneous teams more effective and efficient compared to teams with similar LSs? This was observed for requirements inspections in [12,13,14,15], where inspectors with different LSs found less overlapping defects, improving the overall team effectiveness and efficiency. This finding raised our interest to investigate whether or not this finding also holds for design inspections.

Hypothesis 03: There is no difference in terms of effectiveness when the members of the inspection team have similar or dissimilar LS.

Hypothesis A3: There is a difference in terms of effectiveness when the members of the inspection team have similar or dissimilar LS.

Hypothesis 04: There is no difference in terms of efficiency when the members of the inspection team have similar or dissimilar LS.

Hypothesis A4: There is a difference in terms of efficiency when the members of the inspection team have similar or dissimilar LS.

4 Study Design

We applied the controlled experiment approach [27] to identify and understand relationships between different variables. The experiment comprised the LSs as independent variables and effectiveness and the efficiency as the dependent variables. The design involved a single treatment to be applied by all participants to one object in order to evaluate the impact of the independent variables on the dependent variables. The treatment was conducting a design inspection and the object was a subset of a real requirements document of a large-scale enterprise resource planning system under development and the corresponding design document. However, the independent variable LS cannot be manipulated and random assignment cannot be achieved, which characterizes our study as a quasi-experiment.

4.1 Study Process

The study process includes four steps: (a) training and pre-test prior to the study; (b) experiment preparation; and (c) experiment execution and data collection; and (d) data evaluation, analysis, and reporting.

Training and Pre-Test. During the *training phase* some basic software design concepts and the relevant defect types were revisited. At the end of the training, a pretest was conducted in which participants were asked to find defects in a class and state-chart diagram (seeded with 11 defects). The material included a small description of a toy problem. The participants had to deliver a defect reporting form including defects found and assigned defect types. During the pretest each participants found at least 2 of the seeded defects and based on their results we assumed that they were ready to take part in the experiment.

Experiment Preparation (day 1) and Experiment Execution (day 2). The experiment was conducted in two days. At the first day, participants answered the consent form, the experience characterization form, and the ILS Questionnaire. Thereafter, on the second day, the participants were asked to find the defects in the design document. Note that the requirements document was considered to be correct. Additionally, the defect types were provided in the task description. Based on the provided material the participants were asked to report defects (including time stamps when defects were found). Finally, the participants were asked to complete the follow-up questionnaire on the executed tasks. Note that participants were supervised during the whole experiment (classroom setting) by the research team. Communication

between the participants was not allowed. Note that the inspection tasks was planned to be accomplished within a maximum 2 hour period. However, participants could take breaks (when needed) and could continue their efforts until they felt comfortable with their results within the 2 hours limit. Individual breaks were reported.

Data Evaluation, Analysis, and Reporting. Based on the instruments we were able to characterize the LSs of each participant. One of the researchers analyzed all defect reporting forms to identify which of the real defects each inspector found individually. LS data and identified defects were registered together with the information from the characterization and follow-up questionnaires in a spreadsheet to be used as the basis for data analysis. To process the information regarding teams, the first author created an automated script, able to combine inspectors into all possible combinations of nominal inspection teams, and to calculate the number of unique true defects found by each team. Given the amount of all possible nominal teams, the effort of manually analyzing this data would not have been manageable.

4.2 Inspection Material

The object under inspection includes a scope description and six functional requirements detailed into two use cases. The design document consist of a class diagram, a state-chart diagram and an activity diagram with 36 defects (33 seeded plus 3 additional real defects found during the experiment). Defects, defect types, and corresponding severity levels were defined by the research team in collaboration with inspection experts based on typical defects in this type of application. Table 2 presents the distribution of defects per defect type (left) and defect severity classes (right). Note that we applied the defect taxonomy proposed by Shull [22] for defect classification. This defect taxonomy is applicable for requirements defects and can also be used in context of design documents [25].

Table 2. Defect Types and Severity Levels.

Defect Types		
Type	Quantity	Share
Omission	9	25%
Ambiguous	7	19%
Incorrect fact	14	39%
Extraneous Information	6	17%
Total	36	100%

Defect Severity Levels		
Severity	Quantity	Share
Easy	10	28%
Medium	17	47%
Hard	9	25%
Total	36	100%

The experiment package includes (a) a consent form; (b) an experience characterization questionnaire; (c) the ILS questionnaire; (d) the task descriptions, including a list of defect types to be found (also referred to as fault checklist); (e) the requirements document and the defect seeded design document (both in Portuguese, all other instruments were provided in English); (f) a reference document including the correct UML notation of the diagrams to be inspected (g) a defect reporting form; and (h) a follow-up (experience) questionnaire.

4.3 Subjects

The first experiment trial had a total of 11 participants. They were students of different levels (1 doctoral student, 4 master students, and 6 last semester bachelor students) taking part in a graduate class on experimental software engineering during the second semester of 2015. Based on the characterization form, it was possible to observe that all subjects were familiar with object oriented analysis and design. The overall experience with inspections was low; all of them had only practiced inspections in the context of classroom projects and none of them was familiar to a specific inspection technique.

The second experiment trial was conducted with 26 participants enrolled in a last semester undergraduate course on software quality during the first semester of 2016. Again, all subjects were familiar with object oriented analysis and design and the overall experience with inspections was low. However, this time 6 of the 26 subjects had some experience conducting inspections in industry.

As the goal of our study focuses on novice inspectors, before aggregating the participants of the two trials, we excluded the 6 participants that informed having experience in industry. Moreover, to avoid other potential confounding factors, we also removed graduate students from the final set, ending with 26 candidates representing inexperienced inspectors.

5 Results

When analyzing the results of the participants, we also excluded subjects that found less than 10% of the defects (3 or less), which we considered a minimum to represent a basic understanding of the task, as outliers. A total of 23 out of the 26 candidates found more than 10% of the defects in the design document. The LSs of the participants and their individual effectiveness and efficiency in the design inspection are shown in Table 3.

Table 3. Effectiveness and Efficiency of Individual Inspectors.

ID	*LS*	*Effectiveness*	*Efficiency*	*ID*	*LS*	*Effectiveness*	*Efficiency*
1	Ref-sen-vis-seq	0.22	5.11	13	Ref-int-vis-seq	0.19	5.12
2	Act-sen-vis-glo	0.39	13.33	14	Ref-sen-vis-seq	0.19	7.00
3	Act-sen-vis-glo	0.28	8.57	15	Ref-sen-vis-seq	0.22	6.08
4	Ref-sen-vis-glo	0.25	7.71	16	Act-sen-vis-seq	0.11	3.20
5	Act-sen-vis-glo	0.39	10.00	17	Ref-int-vis-glo	0.11	5.22
6	Ref-sen-vrb-seq	0.11	4.36	18	Ref-sen-vis-glo	0.11	2.50
7	Ref-sen-vis-seq	0.25	5.40	19	Act-sen-vis-seq	0.11	3.58
8	Ref-sen-vis-glo	0.17	6.00	20	Act-sen-vis-glo	0.17	4.00
9	Act-sen-vis-seq	0.11	4.36	21	Act-sen-vis-glo	0.17	4.39
10	Act-int-vrb-glo	0.14	4.29	22	Ref-sen-vis-seq	0.11	3.53
11	Act-sen-vis-seq	0.22	5.78	23	Ref-int-vis-glo	0.22	7.06
12	Act-int-vis-seq	0.19	6.36				

It can be seen that participants found between 11% and 39% of the defects. Hereafter, we use the results of the participants, considering which true defects each one found, to address our research questions.

5. 1 Impact of Learning Styles on Inspection Performance

RQ-1 focuses on the impact of Learning Styles on Inspection Performance: *Do the Learning Styles (LSs) dimensions have an impact on individual defect detection effectiveness and efficiency?*

To answer this question the LSs and number of defect found by participants were analyzed individually. The 23 participants had 9 different LSs and the average effectiveness and efficiency by LS are shown in Fig. 2. This figure also shows the number of subjects in each LS.

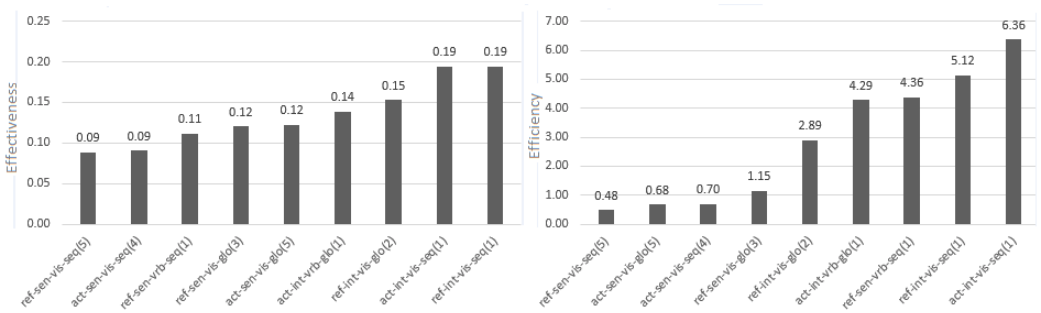

Fig. 2. Effectiveness per LS and Efficiency per LS

The differences of effectiveness and efficiency per LS seem to support the alternative hypotheses A1 and A2. However, analyzing the LS as a whole it is difficult to identify if and how the LS dimensions affecting the results. Therefore, Fig. 3 and Fig. 4 show Boxplots (generated with R Studio) isolating each of the four LS dimensions.

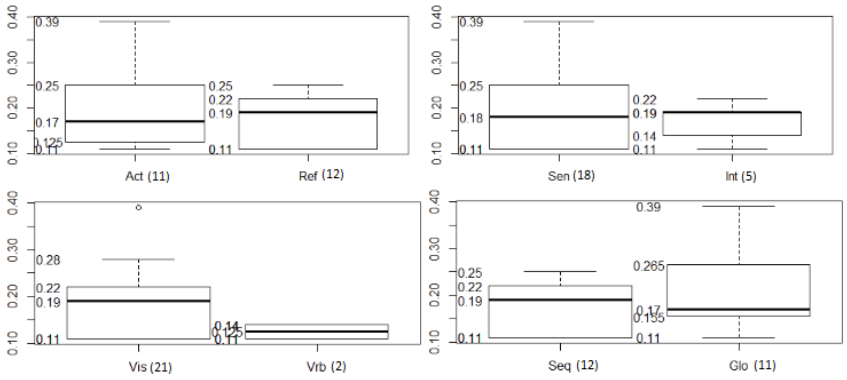

Fig. 3. Effectiveness per LS in each dimension

It can be seen that, isolating the LS dimensions, there are only subtle differences in the median values. To test the null hypotheses H01 and H02, we applied a two-tailed Mann–Whitney–Wilcoxon to calculate the p-values for each of the LS dimensions to verify if one of them affects the effectiveness or efficiency. The resulting p-values are shown in Table 4.

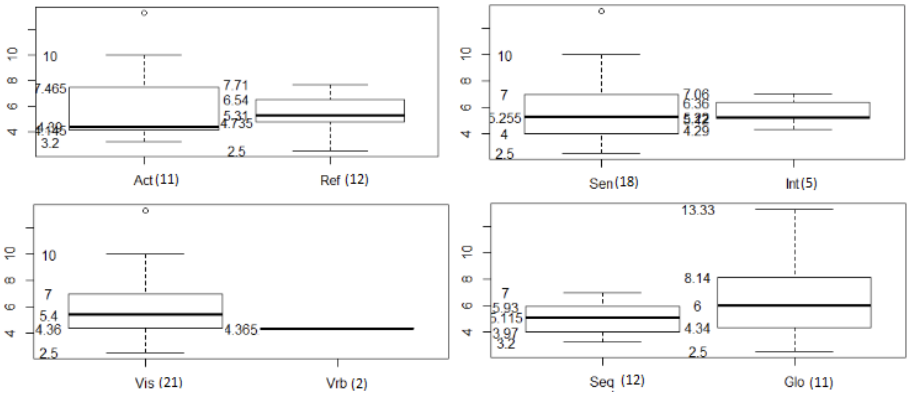

Fig. 4. Efficiency per LS in each dimension

Differently from what has been expected based on Fig. 2, the LS dimensions were not affecting the effectiveness nor the efficiency. Therefore, our data does not allow rejecting the null hypotheses (not even for significance level (α) = 0.2) and the LS dimensions do not seem to have an influence on inspector performance during design inspections. However, although we aggregated data from two different trials we still have a limited amount of subjects and replications are required to improve the conclusion validity.

Table 4. P-values of effectiveness and efficiency per dimension of LS

	Act-Ref	***Sen-Int***	***Vis-Vrb***	***Seq-Glo***
p-value (effectiveness)	0.851	0.704	0.201	0.331
p-value (efficiency)	0.951	0.737	0.275	0.207

5.2 Homogenous versus Heterogeneous Inspection Teams

The second research question (RQ-2) focuses on the heterogeneity of inspection teams: *Are teams with a diversity of LSs more effective and efficient than teams with similar LSs?*

To address RQ-2, the first author created an automated script to generate all possible combinations of nominal teams and counted the unique defects found by each team in order to calculate its effectiveness and efficiency. Thereafter, the teams with similar LS were separated from teams with dissimilar LS for each size as shown in Table 5.

Our similarity criteria were defined as follows. We considered teams as similar if its members have at least two equal LS dimensions (e.g. *act-sen-vis-glo* is similar to *act-sen-vrb-seq* because of *act* and *sen* dimensions). Besides that, we consider that opposite categories in a specific LS dimension are equal when the level is weak (e.g. act[weak] is equal to ref[weak]) and we did not distinguish between strong and moderate levels (e.g. ref [strong] is equal to ref [moderate]).

Table 5. P-values of similar and dissimilar Learning Styles.

Team Size	*Similar Teams*	*Dissimilar Teams*	*p-value (effectiveness)*	*p-value (efficiency)*
2	240	13	**0.002**	**0.016**
3	1312	459	**>0.000**	**>0.000**

The boxplots for effectiveness and efficiency of similar/dissimilar teams per team size (2 and 3) are shown in Fig. 5 and Fig. 6, respectively. Again we applied the Mann-Whitney-Wilcoxon. Based on the p-values (Table 5) we could reject the null hypotheses H03 and H04 for a significance level (α) of 0.05. Nevertheless, it is noteworthy that the differences shown in our results (supporting alternative hypotheses HA3 and HA4) favor the similar treatment. Indeed, observing the boxplots is possible to see that similar teams got better results than dissimilar teams for both sizes in effectiveness and efficiency. The effect size for effectiveness and efficiency of teams of size 2 are 1.03 and 0.82 standard deviations and for size 3 are 0.44 and 0.26, respectively. I.e., for teams of size 2 the average effectiveness of a similar team is more than one standard deviation higher than the average effectiveness of a dissimilar team.

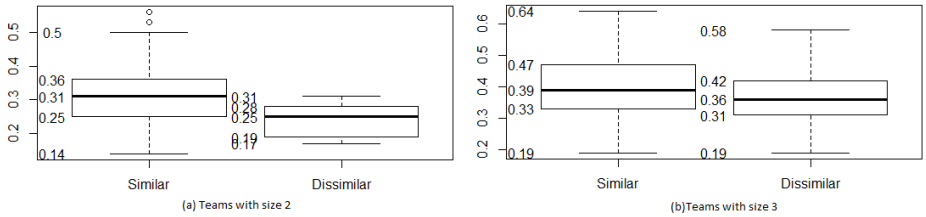

Fig. 5 Effectiveness for similar and dissimilar teams with size 2 and teams with size 3

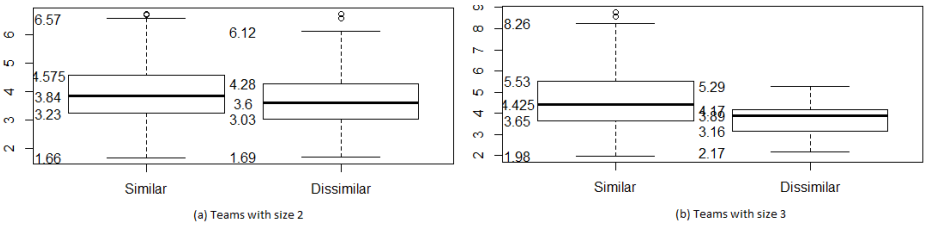

Fig. 6 Efficiency for similar and dissimilar teams with size 2 and teams with size 3

Again our results contradicted the general expectation. In fact, the behavior observed by Goswami et al. [12,13,14,15] for requirements inspections did not occur in our design inspection experiment. For design inspections, the teams with similar LSs were more effective and efficient. This could of course have some influence of our

sample of participants (representing 9 different LS) in which certain LSs are more frequent than others, implying in having more teams with similar LSs. Hence, replications with more participants and different distributions of LSs could again improve validity. Further concerns on threats to validity follow.

6 Threats to Validity

In this section, we identify and discuss the potential threats to validity of our study, organized by the categories described by Wohlin et al. [27], and describe how we addressed them.

Internal Validity. Internal validity concerns a causal relationship between the experiment treatment and the observed results, without the influence of potential confounding factors that are not controlled or measured [27]. The experiment task were conducted individually with the supervision of the researchers. A potential operational difficulty could be related to the fact that the design document included a state-chart diagram, while this diagram was not used in the training and the pretest. However, participants received a summary of the correct notation of the diagram types to be inspected. We also used blocking on our sample to remove some potential confounding factors.

External Validity. External validity refers to the generalization of the results to a larger population or to other environments and to populations that differ from the one studied [27]. Although real and representative requirements and design documents were used, no external validity claims are made throughout the paper and the study is specific to the context in which it was conducted. Also, our participants had a specific distribution of LS. Therefore, replications in other contexts are needed to enable discussing external validity.

Construct Validity. Construct validity concerns the relationship between theory and the observation, i.e., if the treatment correctly reflects the cause construct and if the result correctly reflects the effect construct [27]. Concerning the treatment, it involved applying a design inspection on an object taken from industry, representing real requirements and design documents. The metrics to measure the effect have been used in several inspection studies. A potential threat to construct validity concerns the seeded defects, which could favor a specific LS. Therefore, we seeded defects of different types and with different levels of difficulty.

Conclusion Validity. Finally, conclusion validity (or statistical conclusion validity) regards the relation between the treatment and the results in terms of statistical significance [27]. To support our result analysis, we used R Studio to apply the Mann-Whitney-Wilcoxon statistic test, which is a nonparametric test of null hypothesis for two independent samples. Outliers were removed to avoid influencing the results. The main threat to conclusion validity concerns our sample size and the distribution of LSs between the participants.

7 Conclusion and Future Work

In this paper, we reported on a quasi-experiment to investigate the impact of LSs on the effectiveness and efficiency of design inspections. Therefore, we used a subset of a real requirements document and corresponding design diagrams with seeded defects and characterized each of the participants according to their LS's.

Regarding our results, the first research question (RQ1) concerned the impact of the LSs dimensions on individual defect detection effectiveness and efficiency and the second (RQ2) concerned investigating if teams with a diversity of LSs are more effective and efficient. Differently from what was expected, when investigating RQ1 we were not able to reject the null hypotheses, i.e., in our quasi-experiment none of the dimension had a significant impact on the effectiveness and efficiency. Moreover, the expectations for RQ2 were also not observed, i.e., teams with similar LS showed significantly more effective and efficient.

The results of our study are particularly interesting given that in software engineering literature results which fail to achieve the general expectations are uncommon, for many reasons, including publication bias. However, these kinds of empirical results are extremely important [24]. While we call for replications to improve validity, the main takeaway of our current results is that design inspections can benefit more from applying defect detection techniques (e.g., [25]) than from analyzing the learning styles of the inspectors.

Acknowledgments

Thanks for the support to CAPES, the Brazilian Research Council (CNPq, process number 460627/2014-7), Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development, Austria.

References

1. Aurum A., Petersson H., Wohlin C.: State-of-the-Art: Software Inspection after 25 years. Journal of Software, Testing, Verification and Reliability, 12(3), 133-154 (2002).
2. Basili V., Rombach H.: The tame project: towards improvement-oriented software environments. IEEE Transactions on Software Engineering, 14(6), 758-773 (1988).
3. Biffl S., Gutjahr W.: Influence of team size and defect detection technique on inspection effectiveness. In: IEEE Software Metrics Symposium, Proceedings. Seventh International, pp. 63-75 (2001).
4. Biffl S., Freimut B., Laitenberger O.: Investigating the cost-effectiveness of reinspections in software development. In: 23rd International Conference on Software Engineering, pp. 155-164 (2001).
5. Biffl S., Kalinowski M., Ekaputra F.J., Serral E., Winkler D.: Building empirical software engineering bodies of knowledge with systematic knowledge engineering. Int. Conf. on Software Engineering and Knowledge Engineering (SEKE), pp. 552-559 (2014).
6. Biffl S., Kalinowski M., Rabiser R., Ekaputra F.J., Winkler D.: Systematic Knowledge Engineering: Building Bodies of Knowledge from Published Research. International Journal of Software Engineering and Knowledge Engineering, 24, 1533 – 1571 (2014).

7. Boehm B., Basili V.R.: Software Defect Reduction Top 10 List. IEEE Computer, 34, 135-137 (2001).
8. Chamillard A.T., Karolick D.: Using learning style data in an introductory computer science course. Educational Psychology, 31(1), 291-295 (1999).
9. Fagan M.E.: Design and Code Inspection to Reduce Errors in Program Development. IBM Systems Journal, 15(3), 182-211 (1976).
10. Felder R.M., Silverman L.K.: Learning and teaching styles in engineering education Engineering education. Engineering Education, 78 (7), 674-681 (1988).
11. Felder R. M., Spurlin J.: Applications, reliability and validity of the index of learning styles. International journal of engineering education, 21(1), 103-112 (2005).
12. Goswami A., Walia G.: An empirical study of the effect of learning styles on the faults found during the software requirements inspection. In: International Symposium on Software Reliability Engineering (ISSRE), pp. 330-339 (2013).
13. Goswami A., Walia G., Singh A.: Using learning styles of software professionals to improve their inspection team Performance. In: Int. Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 680-685 (2015).
14. Goswami A., Walia G., Rathod U.: Using Learning Styles to Staff and Improve Software Inspection Team Performance. In: Software Reliability Engineering Workshops (ISSREW), pp. 9-12 (2016)
15. Goswami A., Walia G., McCourt M., Padmanabhan G.: Using Eye Tracking to Investigate Reading Patterns and Learning Styles of Software Requirement Inspectors to Enhance Inspection Team Outcome. In: International Symposium on Empirical Software Engineering and Measurement (ESEM), p. 34 (2016)
16. Johnson P.M.: Does Every Inspection Really Need a Meeting?. Empirical Software Eng. Journal, 3(3), 9-35 (1998).
17. Jung C. G.: Collected Works of CG Jung, Volume 6: Psychological Types. Vol. 6. Princeton University Press (2014)
18. Kalinowski M., Travassos G.H.: A computational framework for supporting software inspections. In: International Conference on Automated Software Engineering (ASE), 46-55 (2004).
19. KoIb, D.A.: Experiential Learning: Experience as the Source of Learning and Development. Prentice-Hall, Englewood Cliffs, N.J. (1984).
20. Montgomery S. M.: Addressing Diverse Learning Styles Through the Use of Multimedia, In: Frontiers in Education Conference, 3a2-13 (1995).
21. Sauer C., Jeffery D.R., Land L., Yetton P.: The Effectiveness of Software Development Technical Review: A Behaviorally Motivated Program of Research. IEEE Transactions on Software Engineering, 26 (1): 1-14 (2000).
22. Shull F.: Developing Techniques for Using Software Documents: A Series of Empirical Studies. PhD Thesis, University of Maryland (1998).
23. Shull F., Rus I., Basili V.: How Perspective-Based Reading Can Improve Requirements Inspections. IEEE Computer, 33 (7), 73-79 (2000).
24. Tichy, W.F.: Hints for Reviewing Empirical Work in Software Engineering. Empirical Software Engineering, 5, 309-312 (2000).
25. Travassos G., Shull F., Fredericks M, Basili V.R.: Detecting Defects in Object Oriented Designs: Using Reading Techniques to Increase Software Quality. In: 14th ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications (OOPSLA), pp. 47-56 (1999).
26. Votta L.: Does Every Inspection Need a Meeting?. ACM Software Eng. Notes, 18(5), 107-114, December (1993).
27. Wohlin C., Runeson P., Höst M., Ohlsson M., Regnell B., Wessl A.: Experimentation in software engineering. Springer (2012).